Social Network (CS522)

Project Report

Bitcoin Alpha Trust Weighted Signed Network

Abstract:

- We used the greedy algorithm approach to solve this.
- Purpose:
    1. We wanted to find the node with the highest degree or the node with maximum number of interactions/connections which also has a high trust rating. This node can be said to be the most influential node in the graph.
    2. WHY?.Once we know that node, we can use that node to further do operations on it. For example, if we use this activity to find the laptop, which has the greatest number of connections. We come to know how to protect this laptop as compared to others or hackers can hack this laptop to get more details. The vulnerability of this node will affect the network/graph most in the event of any attack.
- We use the edited set (removing the last column and sort it according to any one column) and making it a list.
- Using that list and traversing the graph to get the nodes and its number of connections while setting a criteria that only the nodes with a trust rating greater than equal to 5 are to be considered.

- Since in matters of money, people with a lower trust rating will also be attracted to the attacks, we can take a lower rating. But the graph becomes unpresentable and difficult to read.
- Then, at last getting the highest degree node.
- Here, in our code, we found the maximum degree ten nodes.

Introduction

This is a who-trust-whom network of people who trade using Bitcoin on a platform called Bitcoin alpha. Since Bitcoin users are anonymous, there is a need to maintain a record of users' reputation to prevent transactions with fraudulent and risky users. Members of Bitcoin Alpha rate other members in a scale of -10 (total distrust) to +10 (total trust) in steps of 1. This is the first explicit weighted signed directed network available for research.

Dataset statistics	
Nodes	3,783
Edges	24,186
Range of edge weight	-10 to +10
Percentage of positive edges	93%

Data Format (The Bitcoin Alpha Dataset was in the following format which we sort and edited according to our requirement):

SOURCE TARGET RATING TIME

- SOURCE: node id of source, i.e., rater
- TARGET: node id of target, i.e., ratee
- RATING: the source's rating for the target, ranging from -10 to +10 in steps of 1
- TIME: the time of the rating, measured as seconds since Epoch.

In the code, we only required the source node, target node and the rating.

What is the problem addressed?

The problem is to find the node in a graph with the most connections, highest trust rating and if the above two nodes are different, then a top ten nodes which follow the best of both the criterias.

Which publication is the inspiration for this project?

This project was inspired by news articles published on the recent hacking of Twitter accounts of prominent multinational companies and other famous personalities. The hacked accounts were used to post phishing messages asking for money.

What is the new idea for addressing the problem?

The idea is to identify the most trusted and well-connected nodes in the network and monitor their social media accounts for any such activities so that such situations can be dealt with faster.

METHOD:

(The Code Explanation)

We downloaded the data from the “Snap Data set” website (https://snap.stanford.edu/data/soc-sign-bitcoin-alpha.html). Then, saved it under the name "bitcoin.csv" (csv format file). We opened the file and edited it so that we have only the first three columns left with us: SOURCE, TARGET, RATING and added this to the list ‘l’. Then we sorted the list and made a new list ‘g. In the dataset, there are around 1700 nodes which we check and observe that after 50 nodes or so the nodes getting influenced are quite less in number so we limit our graph to 50 for finding the

maximum influential nodes. And the rating we choose is 5 because if we choose a lower rating like 1, 2, 3 or 4, we get a graph which was so dense that one cannot observe anything in that graph and was not presentable. We then store the values in a dictionary with its key as target node and each key has a list corresponding to it, which contains the number of influenced nodes and a set of those influenced nodes.

Then, we again create a new list in which we add dictionary values. This list has two values one is the counter (count of the number of influenced nodes) and the source node (node which is influencing them) for example if the element of list is [35,2] then, 35 is the number of the nodes influenced and 2 is the influencing node. Now, why this orientation? Because when we sort in python, it gets sorted according to the first column element in the list. Why do we convert to list and not use a dictionary? Because dictionary elements can't be addressed by indices.

After this, we check for the maximum connected nodes in the graph one by one for 10 times and store their set of the nodes in a "spset"(superset). We don't want the double nodes so we delete those nodes which are in spset, after finding that one node, from the dictionary and then continue the same process again for ten times. For example, if there are four sets (A, B, C, D) there may be the case that they are having common elements and but we don't want that we get double result so what we can do is we can subtract A from other three sets B, C, D and similarly for the other sets as a result we won't have double elements in our result. Here, we did one more thing: we delete the initial list 'ls' and make a new list 'ls' and append it accordingly. Why? Because it will give greater time complexity if I do the transverse and other operations on the same list.

After this, we did the normal procedure of making a directed graph and plotting the graph with labels and colors: red for main influencing nodes, cyan for spset and yellow for the rest

IMPLEMENTATION:

Libraries used:

1. Networkx
2. Random
3. Matplotlib.pyplot
4. Numpy

RESULT: (Experimental findings)

In the result, our code is mainly displaying the final nodes with maximum connections with trust rating greater than equal to 5. But the result file that we will be sharing contains the contents of the dictionary before changes and spset.

Also, the final figure of the graph.

References

Twitter: 'Embarrassed' Twitter reveals 130 accounts were hacked, 8 profiles downloaded - The Economic Times (indiatimes.com)

BTC-Alpha: Bitcoin Exchange | Cryptocurrency Exchange (btc-alpha.com)

2020 Twitter bitcoin scam - Wikipedia

https://networkx.org/